
# Data Management for the STAR Experiment

Torre Wenaus
BNL

Workshop on Scientific Data Management
10/21/98

# STAR at RHIC

RHIC: Relativistic Heavy Ion Collider at Brookhaven National Laboratory

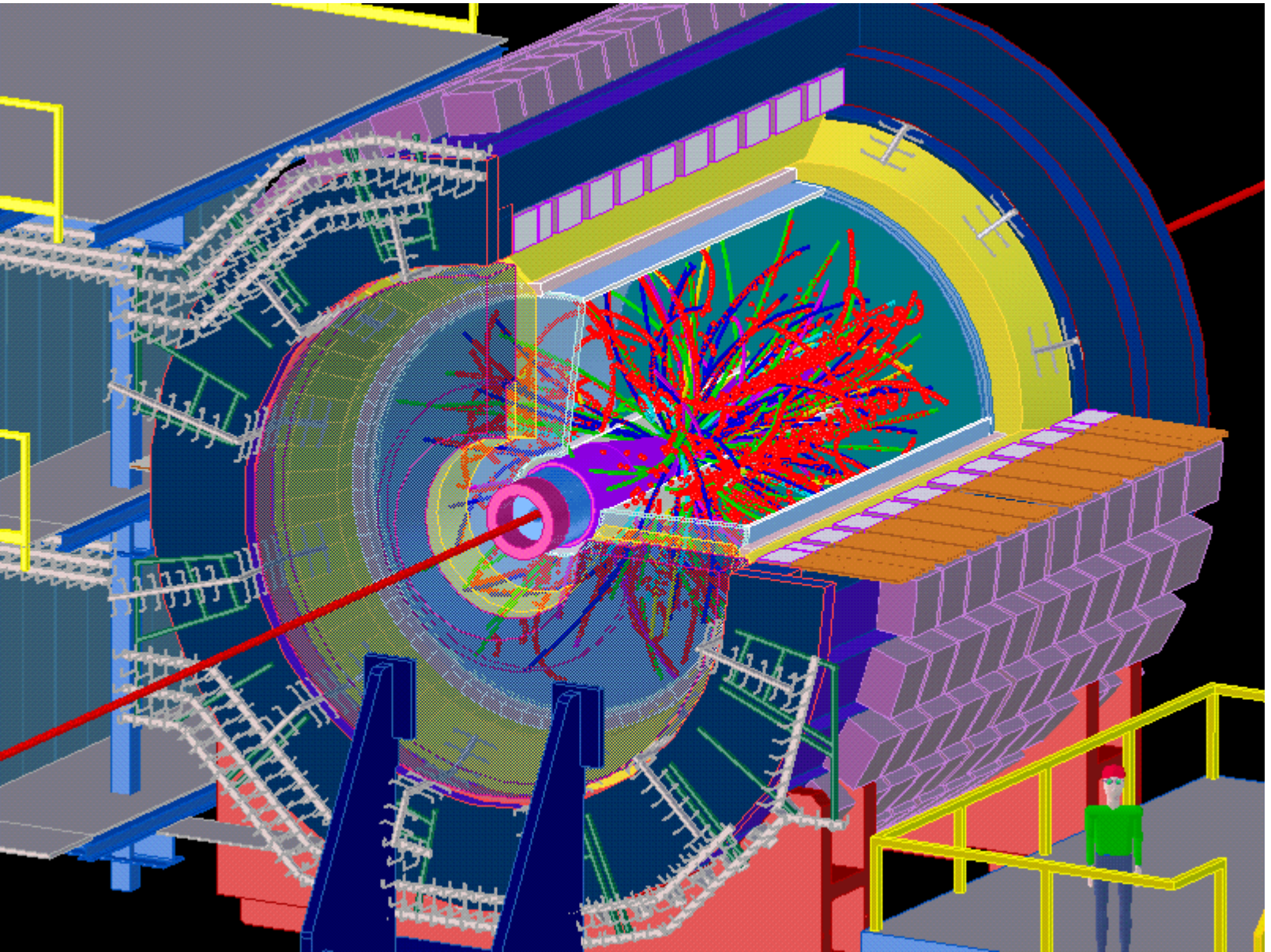
- Colliding Au - Au nuclei at 200GeV/nucleon
- Collision species from p to Au and energies 30-200 AGeV
- Principal objective: Discovery and characterization of the Quark Gluon Plasma
- Additional spin physics program in polarized p - p collisions
- Commissioning run 7/99; first year physics run 11/99

Nuclear physics experiments at a very large scale characteristic of HEP

- Two large experiments, STAR and PHENIX, representing ~80% of the experimental program; >400 collaborators each
- Two smaller, PHOBOS and BRAHMS

STAR experiment

- Heart of experiment is a Time Projection Chamber (TPC) drift chamber together with Si tracker (SVT) and electromagnetic calorimeter
- Precise measurement of hadrons, jets, electrons and photons over large solid angle



# The STAR Computing Task

STAR trigger system reduces 10MHz collision rate to ~1Hz recorded to tape

Data recording rate of 20MB/sec; ~12MB raw data size per event (20MB for simulated events)

- ~4000+ tracks/event recorded in tracking detectors
- High statistics per event permit event by event measurement and correlation of QGP signals such as strangeness enhancement, J/psi attenuation, high Pt parton energy loss modifications in jets, global thermodynamic variables (eg. Pt slope correlated with temperature)
- 17M Au-Au events (equivalent) recorded in nominal year
- Relatively few but highly complex events requiring large processing power
- Wide range of physics studies: ~100 concurrent analyses in ~7 physics working groups

# STAR Computing Requirements

Nominal year processing and data volume requirements:

Raw data volume: 200TB

Reconstruction: 2800 Si95 total CPU, 30TB DST data

- 10x event size reduction from raw to reco assumed (current: 3x)
- 1.5 reconstruction passes/event assumed

Analysis: 4000 Si95 total analysis CPU, 15TB micro-DST data

- 1-1000 Si95-sec/event per MB of DST depending on analysis
  - Wide range, from CPU-limited to I/O limited
- ~100 active analyses, 5 passes per analysis
- micro-DST volumes from .1 to several TB

Simulation: 3300 Si95 total including reconstruction, 24TB

**Total nominal year data volume: 270TB**

**Total nominal year CPU: 10,000 Si95**

# STAR Computing Facilities

Dedicated RHIC computing center at BNL, the RHIC Computing Facility

- Charged with meeting the data archiving and processing needs (reconstruction and analysis; not simulation) of the four experiments
- Three production components: Reconstruction and analysis services (CRS, CAS) and managed data store (MDS)
- 10,000 (CRS) + 7,500 (CAS) Si95 CPU, balanced between CPU-intensive farm processing (reconstruction, some analysis) and I/O intensive SMP processing (I/O intensive analysis)
- ~50TB disk, 270TB robotic tape, 200MB/s, managed by HPSS

Limited resources require the most cost-effective computing possible

- Commodity Intel farms (running Linux) for all but I/O intensive analysis (Sun SMPs)

Support for (a subset of) physics analysis computing at home institutions

# STAR Simulation

Simulation resources must be found outside RCF

- STAR presently utilizes Cray T3Es at NERSC and PSC
- After major effort in last six months, full simulation software base ported to T3Es
  - CERNLIB, Geant3.21, STAR simulation codes and analysis framework, generator codes (HIJING, VENUS)
- Full scale production since August has yielded 2TB of data
  - Ongoing program in support of reconstruction and analysis development
- Current (through Spring '99) needs met; longer term simulation resources unresolved

# STAR Software Environment

Existing software base primarily Fortran

- Will persist into STAR operation
- Not only Fortran but 11k lines of KUMAC scripts (!) and thousands of lines of MORTRAN (HENP folks just felt a cold chill of horror)

Decision taken to migrate to C++/OO

- Existing analysis framework, STAF, is C++/OO foundation supporting both Fortran and C++ applications
  - Migratory approach to incorporating C++
  - Key to manageable migration to C++: data model defined in language-independent IDL
    - Auto-generated code supports access to data model and modular application codes from both languages
    - Recent extension to ROOT supports operation of STAF-based codes and data model access from ROOT without code modification
      - ROOT is a C++/OO analysis tool developed as a successor to PAW by Rene Brun et al. at CERN; http://root.cern.ch

# C++ in STAR

- New projects being undertaken in C++
- Broad interest (some resignation) in going to C++; very little experience
- Training process begun and C++-based tools and environment under development
  - STAR Class Library building on CLHEP (and being incorporated into CLHEP)
  - ROOT-based access to data and data processing codes in use to evaluate scope of deployment of ROOT in STAR
    - ROOT-based production in current Mock Data Challenge
- Existing code is almost entirely in simulation and reconstruction
- Objective:
  - Standardize on C++ based data model and code from post-reconstruction DST through physics analysis
  - Fortran compatibility **not** required or supported in this domain
  - Gradually migrate upstream codes to C++

# Data Management

RHIC Event Store Task Force last fall addressed data management alternatives

- Requirements formulated by the four experiments
- STAR and PHENIX selected Objectivity as the basis for data management
- ROOT selected by the smaller experiments and seen by all as analysis tool with great potential
- Issue for the two larger experiments:
  - Where to draw a dividing line between Objectivity and ROOT in the data model and data processing

# Data Management Requirements

Support C++ with a well documented API.

Must scale to handle data set sizes at RHIC.

Ability to operate in range of the aggregate I/O activity required.

Currently support or plan to support an interface with HPSS

Provide adequate levels of integrity and availability.

Ability to recover from permanently lost data.

Object versioning and schema evolution.

Maintainable and upgradeable. Long-term availability.

Support for read/write access control at the level of individuals and groups.

Administration tools to manage the database.

Backup and recovery of subsets of the data.

Support for copying, moving data; data distribution over WAN

Control over data locality

# Objectivity in STAR

Objectivity selected for all database needs

- Event store
- Conditions and calibrations
- Online database and detector configurations

Decision to use Objectivity would have been inconceivable were it not for BaBar leading the way with an intensive development effort to deploy Objectivity on the same timescale as RHIC and at a similar scale

- From Nov 1 STAR will have *one* dedicated FTE on database development (event store, conditions and configuration)

STAR has imported the full BaBar software distribution and deployed it as an 'external library' providing the foundation for STAR event store and conditions database

- BaBar's software release tool and site customization features used to adapt their code without direct modification to the STAR environment

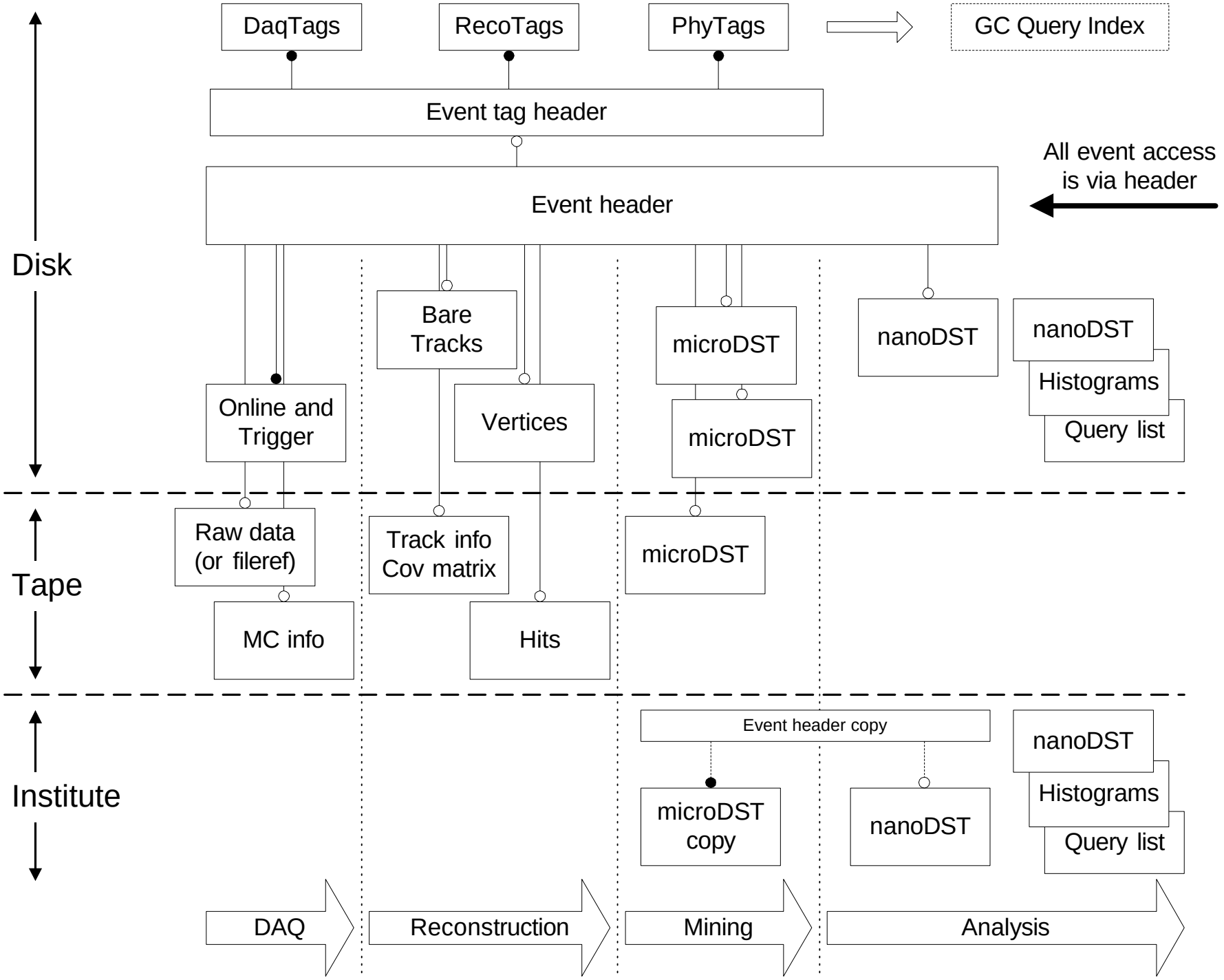
# Objectivity-based Event Store

BaBar event store software adapted to support STAR event store down to the level of event components

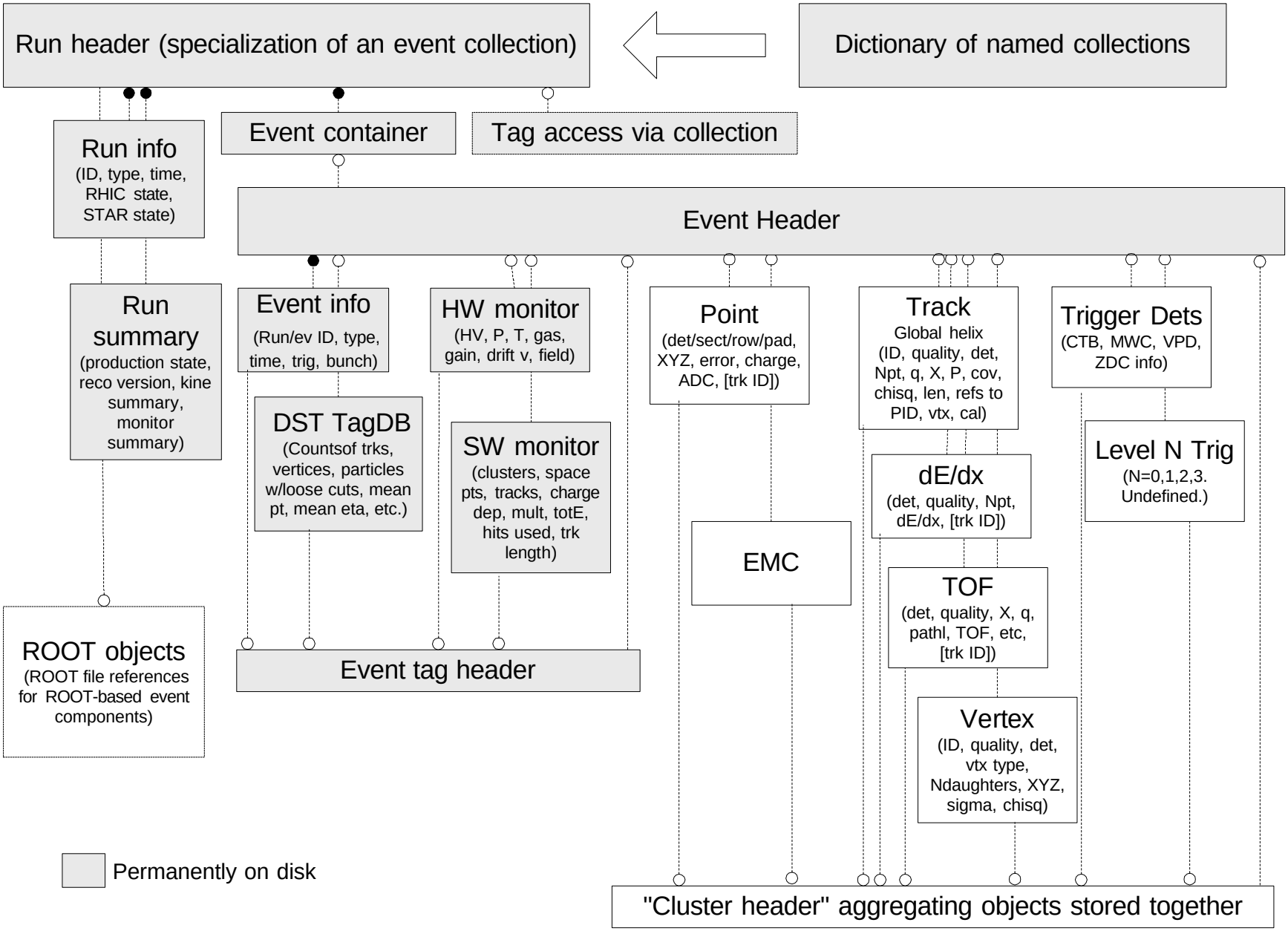
- Event collection dictionary with collection organization in Unix directory tree style
- System, group, user level authorization controls
- Federated database management: developer-level federations, schema evolution
- Event collection implementation
- Event components are purely STAR implementation and map directly to IDL-defined data structures

STAR raw data will **not** be stored in Objectivity

- In-house format will be used to insulate STAR against long-term viability of Objectivity-based storage
- Event store will provide file pointers to the raw data, and an Objectivity-wrapped version of the in-house format for small-scale convenience use


DaqTags
RecoTags
PhyTags
GC Query Index
Event tag header
Event header
All event access is via header
Disk
Bare Tracks
microDST
nanoDST
nanoDST
Histograms
Query list
Online and Trigger
Vertices
microDST
Tape
Raw data (or fileref)
Track info Cov matrix
microDST
MC info
Hits
Institute
Event header copy
nanoDST
Histograms
Query list
microDST copy
nanoDST
DAQ
Reconstruction
Mining
Analysis

# Objectivity-Based Persistent DST Event Model for MDC1



# Conditions and Configuration Databases

Prototype conditions database supporting time-dependent calibrations implemented based on BaBar's conditions database

- Like the event store, builds on existing IDL-based data model

Prototype configuration database supporting STAR's online system developed standalone, for later integration into BaBar framework

# Data Access and Data Mining

Crucial issue in STAR is rapid, efficient data availability for physics analysis

- Full DST and post-DST volume far in excess of disk capacity

A DOE Grand Challenge initiative addresses managed access to and 'mining' of the HPSS-resident Objectivity-based DST data

- Next talk
- Focused presently on RHIC and with heavy STAR involvement
- Very practically addressed to immediate RHIC needs
- Already a production-capable tool being integrated into current STAR data management regulating access to HPSS-resident DSTs

# Mock Data Challenge (MDC)

RHIC-wide exercise in stress-testing RCF and experiment hardware and software in full production environment from (simulated) raw data through physics analysis

MDC1 just concluded, with objective of processing (at least) 100k events per experiment on 10%-nominal RCF facilities

A success for RHIC and STAR

In STAR,

- 217k Au-Au events (1.7TB) simulated on the T3Es with network transport to RCF HPSS
- 168k events reconstructed (600GB of DSTs generated)
- 50GB Objectivity DST database built (disk space limited) and used at a small scale with STAR analysis codes
- Full 600GB HPSS-resident DST database will be built in ~2 weeks using Grand Challenge software to manage access from STAR analysis codes

# Event Store Implementation and Status

STAF-standard XDF format (based on XDR) for IDL-defined data is today the production standard and will remain an Objectivity fall-back

- IDL-defined persistent data model ensures XDF compatibility
- IDL-standard restricts how we use Objectivity: don't even pretend it's C++
- But, STAR (taking a lesson from BaBar) is completely decoupling persistent and transient data models, bearing the cost of translation
  - Design of C++ transient model is decoupled from persistent implemenation
  - Transient representation is built in the translator
  - No direct exposure of application code to Objectivity

Objectivity/BaBar based event store deployed in current Mock Data Challenge and now a production fixture of STAR data processing

Linux support is a crucial issue; currently limited to Sun/Solaris

- With Objectivity port available, BaBar/STAR software porting will proceed by end of year

# A Hybrid Event Store for STAR?

Requirements driving STAR to Objectivity are grounded in the very large scale data management problem

- We're comfortable, so far, with Objectivity in the global data management role

Requirements and priorities for select components of the data model differ and can drive different choices

- Non-Objectivity raw data already addressed

Post-DST physics analysis data (micro-DSTs) is such an area

- High value in close coupling to data analysis tool (ROOT, in all probability): direct access to data model during analysis, presentation
- Great physicist-level flexibility essential in defining object model (schema), and Objectivity (currently) presents severe problems in secure, flexible schema management
- Premium on storage optimization for compact, rapid-access data (compression, N-tuple like storage)

# Hybrid ROOT/Objectivity Event Store

These considerations motivate the use of ROOT for micro-DST level persistent data

- Particularly given the immaturity of the analysis tools being developed for LHC to work in conjunction with Objectivity-based data

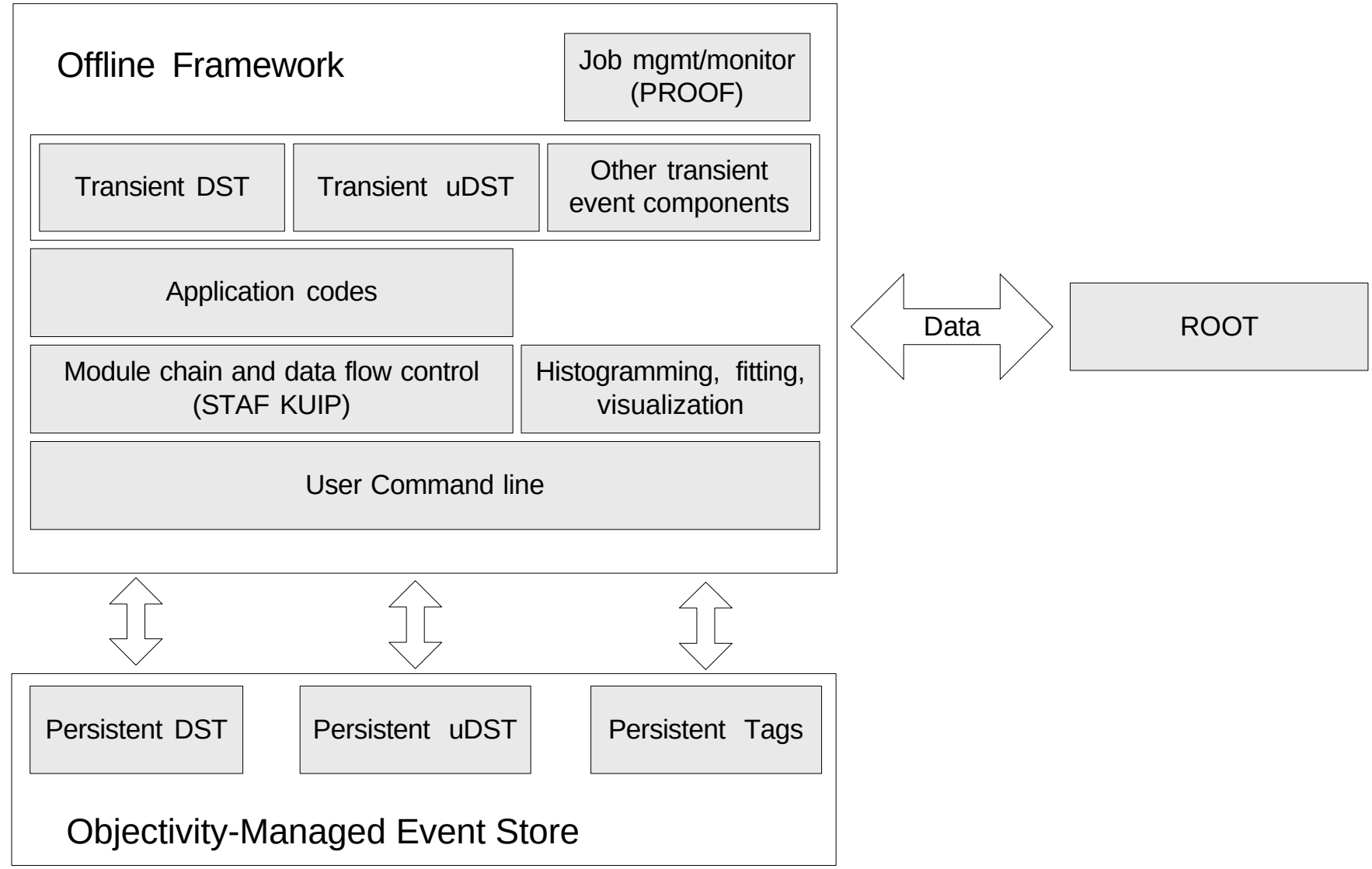
STAR is currently investigating ROOT-based micro-DSTs integrated into the Objectivity event store

- ROOT-based micro-DSTs associated with an event collection incorporated via ROOT file references at the collection level

# "Using ROOT" is many decisions, not one

Root can occupy or be used directly in any component...

... or none of them, used only as an external visualization/analysis tool



# Conclusions

The circumstances of STAR

- Startup in 1999
- Slow start in addressing event store implementation, C++ migration
- Large base of legacy software
- Extremely limited manpower and computing resources

drive us to extremely practical and pragmatic data management choices

- Beg, steal and borrow from the community
- Deploy community and industry standard technologies to leverage future as well as current work elsewhere
- Isolate implementation choices behind standard interfaces, to revisit and re-optimize in the future

which leverage existing STAR strengths

- Component and standards-based software greatly eases integration of new technologies
  - preserving compatibility with existing tools for selective and fall-back use
  - while efficiently migrating legacy software and legacy physicists

By today's judgement, data management is on track to have a capable system by startup that scales to STAR's data volume and 10-15yr lifespan.